

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Application of:	§	
JHA, et al.	§	Confirmation No.: 4536
Serial No.: 10/731,383	§	Group Art Unit: 2145
	§	
Filed: December 9, 2003	§	Examiner: Lin Liu
For: TRANSMITTING COMMANDS AND	§	
INFORMATION BETWEEN TCP/IP	§	
STACK AND AN OFFLOAD UNIT		

**MAIL STOP APPEAL BRIEF-PATENTS**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

Dear Sir:

Applicant submits this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2145 dated January 18, 2008, finally rejecting claims 1-20 (the "Final Office Action"). The final rejection of claims 1-20 is appealed. This Appeal Brief is believed to be timely since it is electronically transmitted by the due date of June 17, 2008, as set by the filing of a Notice of Appeal on April 17, 2008. The Commissioner is hereby authorized to charge Deposit Account No. 20-0782/NVDA/P000862/SW for any fees necessary to make this Appeal Brief timely and acceptable to the Office.

## TABLE OF CONTENTS

1.	Identification Page.....	1
2.	Table of Contents .....	2
3.	Real Party in Interest .....	3
4.	Related Appeals and Interferences .....	4
5.	Status of Claims .....	5
6.	Status of Amendments .....	6
7.	Summary of Claimed Subject Matter .....	7
8.	Grounds of Rejection to be Reviewed on Appeal .....	9
9.	Arguments .....	10
10.	Conclusion .....	14
11.	Claims Appendix .....	15
12.	Evidence Appendix .....	22
13.	Related Proceedings Appendix .....	23

### **Real Party in Interest**

The present real party in interest is NVIDIA Corporation, a corporation of the State of Delaware, having a place of business at 2701 San Tomas Expressway, Santa Clara, CA 95050.

### **Related Appeals and Interferences**

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

## **Status of Claims**

Claims 1-27 and 32-41 are pending in the application. Claims 1-31 were originally presented in the application. Claims 1-27 and 32-41 stand finally rejected as discussed below. The final rejections of claims 1-27 and 32-41 are appealed. The pending claims are shown in the attached Claims Appendix.

## **Status of Amendments**

All claim amendments prior to and including the Final Office Action have been entered by the Examiner.

### Summary of Claimed Subject Matter

Claimed embodiments include a method of communicating between a TCP stack and an offload unit using a command ring (*see, e.g.*, claims 1-6, 32-35, 37, and 39-41), a method of communicating between a TCP stack and an offload unit using a notification ring (*see, e.g.*, claims 7-15), a system for transmitting commands from a TCP stack to an offload unit using a command ring (*see, e.g.*, claims 16-21), and a system for transmitting connection information from an offload unit to a TCP stack using a notification ring and a command ring (*see, e.g.*, claims 22-27 and 38).

Claim 1 recites a command ring that is used to transfer commands from the TCP stack to the offload unit and command specific status from the offload unit to the TCP stack. As claimed, the method includes the steps of: indicating by a bit an owner of an entry in the command ring, the owner being the offload unit; reading a command from an entry in the command ring; setting the bit in the entry by the offload unit to indicate the owner of the entry is the TCP stack; and writing command specific status to the entry, *Application*, paragraph [0070] and Figure 6A. An important aspect of Appellants' invention is a bit that indicates an owner of each one of the command ring entries is included in every entry of the command ring.

Claim 7 recites a notification ring that is used to transfer notification descriptors from the offload unit to the TCP stack. As claimed, the method includes the steps of: writing a notification descriptor to an entry in a notification ring by the offload unit; indicating by a bit an owner of the entry, the owner being the TCP stack; reading the notification descriptor from the entry in the notification ring, *Application*, paragraph [0078] and Figure 6B. An important aspect of Appellants' invention is a bit that indicates an owner of each one of the notification ring entries is included in every entry of the notification ring.

Claim 16 recites a command ring that is used to transfer commands from the TCP stack to the offload unit. As claimed, a system includes: a command ring configured to receive commands written by the TCP stack; a driver configured to write a command to an entry in the command ring; the driver, indicating by a bit the offload unit as an owner of the entry; and the offload unit configured to read the entries in the

command ring that indicate the offload unit as the owner, process the commands, and set the bit indicating that the TCP stack is the owner in any entries that are read by the offload unit, *Application*, paragraph [0070] and Figure 6A. An important aspect of Appellants' invention is a bit that indicates an owner of each one of the command ring entries is included in every entry of the command ring.

Claim 22 recites a command ring that is used to transfer commands from the TCP stack to the offload unit and a notification ring that is used to transfer notification descriptors from the offload unit to the TCP stack. As claimed, a system includes: a notification unit within an offload unit configured to write notification descriptors to entries in a notification ring; indicating by a bit an owner of each entry, the owner being the TCP stack; the notification ring configured to output the notification descriptors read by the TCP stack from the entries in the notification ring that indicate the TCP stack as the owner, and the TCP stack configured to process the notification descriptors and set the bit indicating that the offload unit is the owner in any entries that are read by the TCP stack, *Application*, paragraph [0078] and Figure 6B. As claimed, the system also includes: a driver configured to write a command to an entry in a command ring; and the TCP stack configured to indicate by a bit an owner of each entry of the command ring that is written by the TCP stack as the offload unit, *Application*, paragraph [0070] and Figure 6A. An important aspect of Appellants' invention is a bit that indicates an owner of each one of the notification ring entries is included in every entry of the notification ring and that a bit that indicates an owner of each one of the command ring entries is included in every entry of the command ring.



### **Grounds of Rejection to be Reviewed on Appeal**

1. Claims 1-3, 5-10, 13-25, 27, 32-38, and 40 are rejected under 35 USC §103(a) as unpatentable over Pinkerton (US Pub. No. 2006/0069792) in view of Boyd (US Pub. No. 2004/0049601) and Lanteigne (US 6,757,756).
2. Claims 4, 39, and 41 are rejected under 35 USC §103(a) as unpatentable over Pinkerton in view of Boyd and Lanteigne and further in view of Boucher (US 6,436,620).
3. Claims 11, 12 and 26 stand rejected under 35 USC §103(a) as unpatentable over Pinkerton in view of Boyd and Lanteigne and further in view of Meyer (US Pub. No. 2002/0145976).

## ARGUMENT

### **Obviousness of Claims 1-3, 5-10, 13-25, 27, 32-38, and 40 over Pinkerton in view of Boyd and Lanteigne**

The Examiner has rejected claims 1-3, 5-10, 13-25, 27, 32-38, and 40 are rejected under 35 USC §103(a) as unpatentable over Pinkerton (US Pub. No. 2006/0069792) in view of Boyd (US Pub. No. 2004/0049601) and Lanteigne (US Patent No. 6,757,756). Appellants respectfully disagree with these rejections.

Claims 1, 7, 16, and 22 each recite the limitation of a bit in every entry of the (command or notification) ring that indicates an owner of each one of the entries. Claim 1 also recites the limitation of setting the bit in the entry by the offload unit to indicate that the owner of entry is the TCP stack. Additionally, claim 7 recites the limitation of the offload unit writing to the notification ring and indicating that the TCP stack is the owner of the entry that is written. Similarly, claim 16 recites the limitation of the offload unit setting the bit indicating the TCP stack is the owner in any entries that are read by the offload unit. Finally, claim 22 recites the limitations of an offload unit that indicates, via a bit, an owner of each entry written by the notification unit as the TCP stack, the TCP stack setting the bit indicating that the offload unit is the owner in any entries in the notification ring that are read by the TCP stack, and the TCP stack indicating, via a bit, an owner of each entry of a command ring that is written by the TCP stack as the offload unit.

The Pinkerton reference does not teach storing a bit in each entry of a command or notification ring, which is a primary feature recited in claims 1, 7, 16, and 22, for passing commands and notification descriptors between the TCP stack and the offload unit. Pinkerton teaches that the offload unit produces a linked list of parameters and dispatch tables (see paragraph [0052]) that is inserted in a completion message. The parameters may be used to provide communication channels with the offload unit. However, there is no ownership associated with the parameters, dispatch tables, or communication channels. Nowhere does Pinkerton teach or suggest indicating ownership of any of the entries in the linked list.

Pinkerton teaches that a DELEGATED variable may be owned by the host or offload unit (see paragraph [0041]). However, the DELEGATED variables are written by the host to the offload unit and read back from the offload unit by the host. The DELEGATED variables are not passed between the host and offload unit through a command or notification ring. Importantly, ownership of the DELEGATED variables is changed from the host to the offload unit when a connection is offloaded, and ownership only reverts to the host when offloading of the connection is terminated. The ownership of the DELEGATED variable is not set to the TCP stack. Pinkerton fails to teach or suggest that a bit in an entry in a command or notification ring is set to indicate the owner of the entry is the TCP stack, as recited in claims 1, 7, 16, and 22.

To provide a teaching of the command ring, the Examiner cites Boyd at page 9, paragraph [0118]. However, this section, and the rest of the reference in its entirety, only teaches the existence of a circular linked list that stores pointers. A ULP monitors the entries in the linked list and uses a credit-based mechanism to control reading from and writing to the linked list. Nowhere does Boyd teach or suggest indicating ownership within each entry of the circular linked list. Like Pinkerton, Boyd also fails to teach or suggest that a bit in an entry in a command or notification ring is set to indicate the owner of the entry is the TCP stack, as recited in claims 1, 7, 16, and 22.

To provide a teaching of the bit in each entry that indicates an owner of each entry, the Examiner cites Lanteigne, relying on the teaching of read and write pointers of a control ring buffer that are owned by processes. The Examiner concludes that it would have been obvious to indicate ownership of each entry using the write and read pointers. A review of the Lanteigne reference, specifically column 9, lines 30-36, show that the read pointer and the write pointer are stored in Master Control Block 706 (see Figure 9). The control ring entries are stored in Control Ring 708. Therefore, neither the read pointer nor the write pointer qualifies as a bit in an entry of the command or notification ring that indicates ownership of the entry, as recited in claims 1, 7, 16, and 22. Furthermore, Lanteigne teaches that a process must “gain ownership” of the read or write pointer in order to access an entry (see column 13, lines 47-59). After accessing the entry, the process “gives up ownership” to permit another process to access an entry of the control ring buffer. Importantly, the processes do not set a bit in

an entry indicating that another process has ownership of an entry. Like Pinkerton, Lanteigne also fails to teach or suggest that a bit in an entry in a command or notification ring is set to indicate the owner of the entry is the TCP stack, as recited in claims 1, 7, 16, and 22.

Claim 2 recites the limitation that the command includes a location of a buffer. As argued by the Examiner on page 7 of the Final Office Action, page 5, paragraph [0043] of Pinkerton teaches this limitation. Pinkerton teaches that data packets are buffered when they cannot be processed. Nowhere does Pinkerton teach or suggest that the command includes a location of a buffer.

Claim 13 recites the limitation that a flag indicates a sequence number threshold was reached. As argued by the Examiner on page 10 of the Final Office Action, page 7, paragraph [0061] of Pinkerton teaches this limitation. Pinkerton teaches that a variable may include, “a flag to turn off Nagling” and “a flag to indicate that Keep-Alive is needed.” Neither of these flags indicate that a sequence number threshold was reached, as recited in claim 13. Pinkerton also teaches that a variable may include various sequence numbers. However, Pinkerton fails to teach or suggest that any of these sequence numbers has a corresponding flag. Finally, Pinkerton also teaches that the variable may include a “slow start threshold.” Although the sequence numbers are listed in the same paragraph of Pinkerton, there is no teaching or suggestion that the slow start threshold is associated with the sequence number and a flag. Nowhere does Pinkerton teach or suggest a flag that indicates a sequence number threshold was reached.

**Obviousness of Claims 4, 39, and 41 over Pinkerton in view of Boyd, Lanteigne, and Boucher and Claims 11, 12 and 26 over Pinkerton in view of Boyd, Lanteigne, and Meyer**

Other references cited by the Examiner include Boucher and Meyer. These other references fail to cure the deficiencies of Pinkerton, Boyd, and Lanteigne relative to claims 1, 7, 16, and 22. Meyer is relied on for teaching a flag that indicates an acknowledgement threshold is reached. Meyer teaches a method of triggering retransmission of data segments when duplicate acknowledgements are received. Boucher is relied on for teaching a value representing a number of buffers that are

accepted by the offload unit, a synchronization pointer, and flushing unused buffer descriptors. Boucher teaches that entries are stored into a receive buffer queue (see column 13, lines 8-26), but does not teach or suggest that a value representing a number of buffers accepted by the offload unit is written into an entry in the command ring, as recited in claim 4. Boucher also teaches synchronization of data passed across a clock boundary. A data synchronization read pointer and a data synchronization write pointer are used to transfer the data across the clock boundary. These data synchronization buffer pointers are not included in a command and do not enable the offload unit to accept user buffer descriptors, as recited in claims 39 and 41 of the present application. Additionally, neither Boucher nor Meyer teach or suggest the limitation of a bit in an entry in a command or notification ring is set to indicate the owner of the entry is the TCP stack, as recited in claims 1, 7, 16, and 22.

As the foregoing illustrates, no combination of the references cited by the Examiner can render independent claims 1, 7, 16, and 22 obvious. Further, since claims 2-6 and 32-35, 36-37, 39-41 depend from allowable claim 1, these claims are also patentable over any combination of the references cited by the Examiner. Claims 8-15, 17-21, and 23-27 and 38 depend from allowable claims 7, 16, and 22, respectively, and are therefore also patentable over any combination of the references cited by the Examiner.

All of the claims currently pending in the application are therefore patentable over Hayes. In view of these clear distinctions, reconsideration and allowance of all the claims is respectfully requested.

## CONCLUSION

The Examiner errs in finding that:

- Claims 1-3, 5-10, 13-25, 27, 32-38, and 40 are unpatentable over Pinkerton in view of Boyd and Lanteigne.
- Claims 4, 39, and 41 are unpatentable over Pinkerton in view of Boyd and Lanteigne and further in view of Boucher.
- Claims 11, 12 and 26 are unpatentable over Pinkerton in view of Boyd and Lanteigne and further in view of Meyer.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Stephanie Winner', is written over a horizontal line.

Stephanie Winner  
Registration No. 52,371  
PATTERSON & SHERIDAN, L.L.P.  
3040 Post Oak Blvd. Suite 1500  
Houston, TX 77056  
Telephone: (650) 330-2310  
Facsimile: (650) 330-2314  
Agent for Applicant

## CLAIMS APPENDIX

1. (Previously Presented): A method of communicating between a TCP stack, wherein the TCP stack delegates one or more connections to the offload unit, and the TCP stack processes connections that are not delegated or require special processing, and an offload unit, comprising:

utilizing a driver as a translator for writing a command including an index corresponding to a delegated connection to an entry in a command ring;

indicating by a bit an owner of the entry, the owner being the offload unit, wherein every entry in the ring includes a bit indicating the owner of the entry;

reading the command from the entry in the command ring to the offload unit;

setting the bit in the entry by the offload unit to indicate the owner of entry is the TCP stack;

executing the command; and

writing command specific status to the entry in the command ring by the offload unit.

2. (Original): The method of claim 1, wherein the command includes a location of a buffer for storing payload data produced by the offload unit.

3. (Original): The method of claim 1, wherein the command includes connection information needed to setup a delegated connection.

4. (Original): The method of claim 1, wherein the command specific status includes a value representing a number of buffers accepted by the offload unit.

5. (Original): The method of claim 1, further comprising:

writing a notification descriptor including an index corresponding to a delegated connection to an entry in a notification ring; and

reading the notification descriptor from the entry in the notification ring.

6. (Original): The method of claim 5, wherein the notification descriptor includes one or more notification flags indicating specific information for a connection.
7. (Previously Presented): A method of communicating between a TCP stack and an offload unit, wherein the TCP stack delegates one or more connections to the offload unit, the TCP stack processing connections that are not delegated or require special processing comprising:
  - writing a notification descriptor to an entry in a notification ring by the offload unit;
  - indicating by a bit an owner of the entry, the owner being the TCP stack, wherein every entry in the ring includes a bit indicating the owner of the entry;
  - utilizing a driver as a translator for reading the notification descriptor from the entry in the notification ring;
  - reading the notification descriptor from the entry in the notification ring to the TCP stack based on the bit; and
  - determining connection information for a delegated connection based on the notification descriptor.
8. (Original): The method of claim 7, wherein the notification descriptor includes an index corresponding to the delegated connection.
9. (Previously Presented): The method of claim 7, wherein the notification descriptor includes one or more notification flags corresponding to the delegated connection.
10. (Original): The method of claim 7, wherein the notification descriptor includes a count of received acknowledgements.
11. (Original): The method of claim 8, wherein a flag indicates an acknowledgement threshold was reached on the delegated connection.
12. (Original): The method of claim 8, wherein a flag indicates a duplicate acknowledgement was received on the delegated connection.



13. (Original): The method of claim 8, wherein a flag indicates a sequence number threshold was reached on the delegated connection.

14. (Original): The method of claim 8, wherein a flag indicates at least a portion of frame data received on the delegated connection was uploaded by the offload unit to a legacy buffer.

15. (Original): The method of claim 8, wherein a flag indicates a request for a user buffer for uploading of payload data from the offload unit.

16. (Previously Presented): A system for transmitting commands from a TCP stack wherein the TCP stack delegates one or more connections to the offload unit, the TCP stack processing connections that are not delegated or require special processing, to an offload unit, comprising:

- a command ring configured to receive commands written by the TCP stack;
- a driver configured as a translator for writing a command including an index corresponding to a delegated connection to an entry in the command ring;

- the driver indicating by a bit, the offload unit as an owner of the entry that is written with the command, wherein every entry in the command ring includes a bit indicating the owner of the entry; and

- the offload unit configured to read entries from the command ring that indicate the offload unit as the owner and to process the commands for the one or more connections delegated to the offload unit and set the bit indicating that the TCP stack is the owner in any entries that are read by the offload unit.

17. (Original): The system of claim 16, further comprising:

- a notification ring configured to receive connection information written by the offload unit and output connection information read by the TCP stack.

18. (Original): The system of claim 16, wherein the offload unit is configured to write command specific status to the command ring.

19. (Original): The system of claim 16, further comprising a transmit descriptor ring configured to transfer transmit buffer information from the TCP stack to the offload unit.

20. (Original): The system of claim 19, wherein the transmit buffer information includes a delegated connection index.

21. (Original): The system of claim 16, further comprising a receive descriptor ring configured to transfer receive buffer information from the TCP stack to the offload unit.

22. (Previously Presented): A system for transmitting connection information from an offload unit to a TCP stack wherein the TCP stack delegates one or more connections to the offload unit, the TCP stack processing connections that are not delegated or require special processing, comprising:

- a notification unit within the offload unit configured to write notification descriptors to entries in a notification ring;

- the offload unit indicating by a bit an owner of each entry written by the notification unit as the TCP stack, wherein every entry in the notification ring includes a bit indicating the owner of the entry;

- a driver configured as a translator for writing a command including an index corresponding to a delegated connection to an entry in a command ring;

- the TCP stack indicating by a bit an owner of each entry of the command ring that is written by the TCP stack as the offload unit, wherein every entry in the command ring includes a bit indicating the owner of the entry;

- the notification ring configured to output notification descriptors read by the TCP stack from the entries that indicate the TCP stack as the owner; and

- the TCP stack configured to process the notification descriptors for the one or more connections that are delegated to the offload unit and set the bit indicating that the

offload unit is the owner in any entries in the notification ring that are read by the TCP stack.

23. (Original): The system of claim 22, wherein the notification unit is configured to set one or more flags, each flag indicating connection information.

24. (Previously Presented): The system of claim 22, wherein the notification unit is configured to set a flag when at least a portion of a partially parsed frame is uploaded to a legacy buffer.

25. (Previously Presented): The system of claim 22, wherein each notification descriptor includes a sequence number.

26. (Previously Presented): The system of claim 22, wherein each notification descriptor includes an acknowledgement number.

27. (Previously Presented): The system of claim 22, wherein each notification descriptor includes a count of received acknowledgements.

28-31. (Cancelled)

32. (Previously Presented): The method of claim 1, further comprising:  
    writing the entry to the command ring by the TCP stack; and  
    setting the bit in the entry comprising Command Write Pointer to indicate that the entry is owned by the offload unit.

33. (Previously Presented): The method of claim 1, wherein when the offload unit writes the entry to the command ring, the bit in the entry comprising command read pointer is set to indicate that the entry is owned by the TCP stack.

34. (Previously Presented): The method of claim 33, wherein when the offload unit writes the entry to the command unit, the bit in the entry comprising Command Read Pointer is set to indicate that the entry is owned by the TCP stack.

35. (Previously Presented): The method of claim 31, wherein at startup the bit indicates the entries owned by the TCP stack.

36. (Cancelled)

37. (Previously Presented): The method of claim 36, including the step of the offload unit using the ring to notify the TCP stack of offloaded connections needing further processing by the TCP stack.

38. (Previously Presented): The system of claim 22, wherein the offload unit is configured to process the commands for the one or more connections that are delegated to the offload unit and set the bit indicating that the TCP stack unit is the owner in any entries in the command ring that are read by the offload unit.

39. (Previously Presented): The method of claim 2, wherein the command includes a synchronization bit configured to enable the offload unit to accept user buffer descriptors that specify locations of buffers for storing payload data produced by the offload unit.

40. (Previously Presented): The method of claim 5, further comprising setting a bit in the entry in the notification ring by the offload unit indicating that the entry is owned by the TCP stack when the notification descriptor is written.

41. (Previously Presented): The method of claim 5, further comprising:  
    setting a synchronization request flag in the notification descriptor; and

flushing unused user buffer descriptors queued in the offload unit for the delegated connection, wherein the unused buffer descriptors specify locations of buffers for storing payload data produced by the offload unit.

## EVIDENCE APPENDIX

None

## RELATED PROCEEDINGS APPENDIX

None